Improvements to the IBM HUB5E System

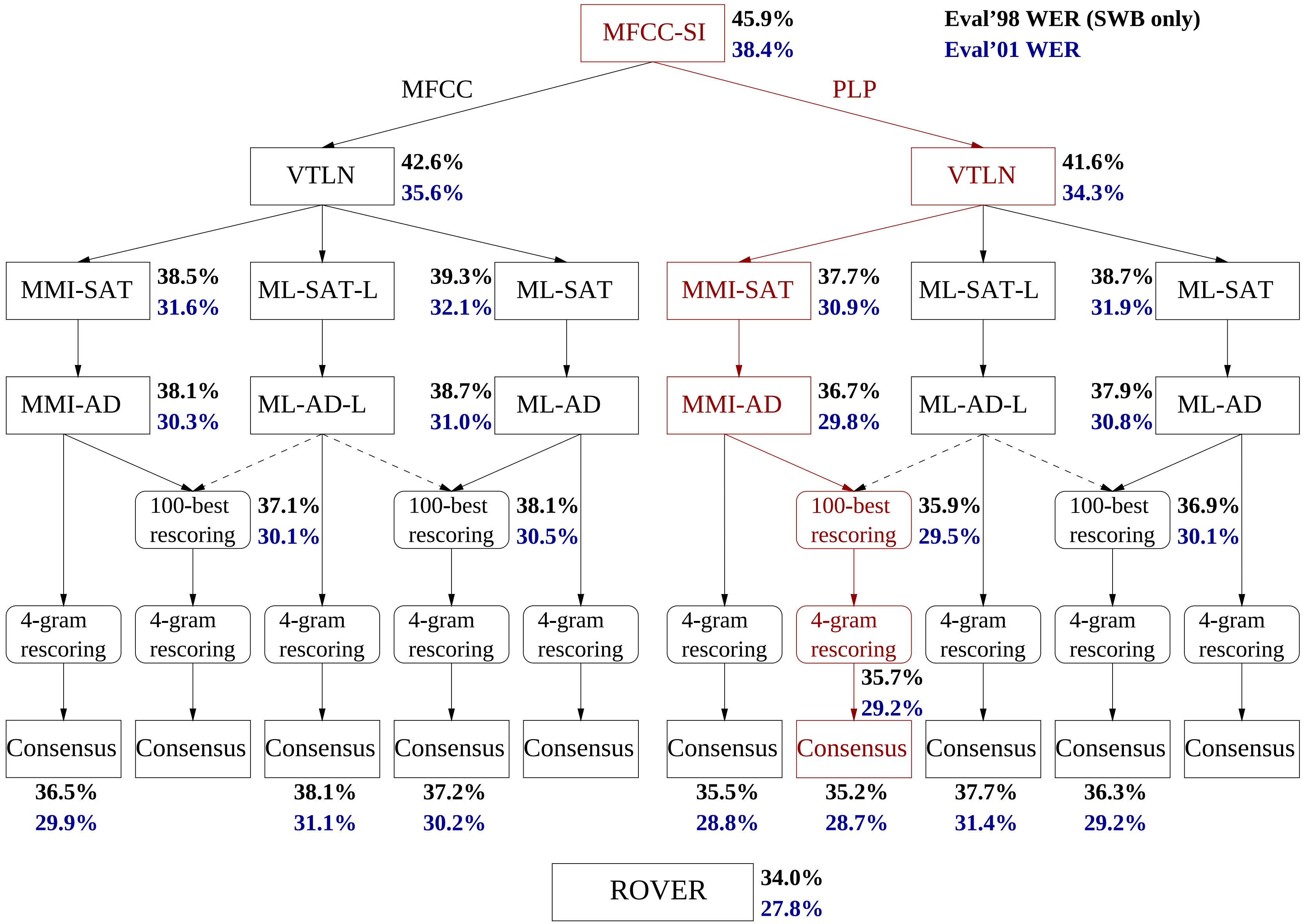
Jing Huang, Brian Kingsbury, Lidia Mangu, George Saon, Geoffrey Zweig

- Michael Picheny
- Peder Olsen, Ramesh Gopinath, Vaibhava Goel, Karthik Visweswariah

Outline

- Last year's evaluation system
- Current system
- Distribution function matching adaptation
- Extended maximum likelihood linear transform (EMLLT)
- Implicit lattice MMI training
- Conclusion

Last year's evaluation system



Current system

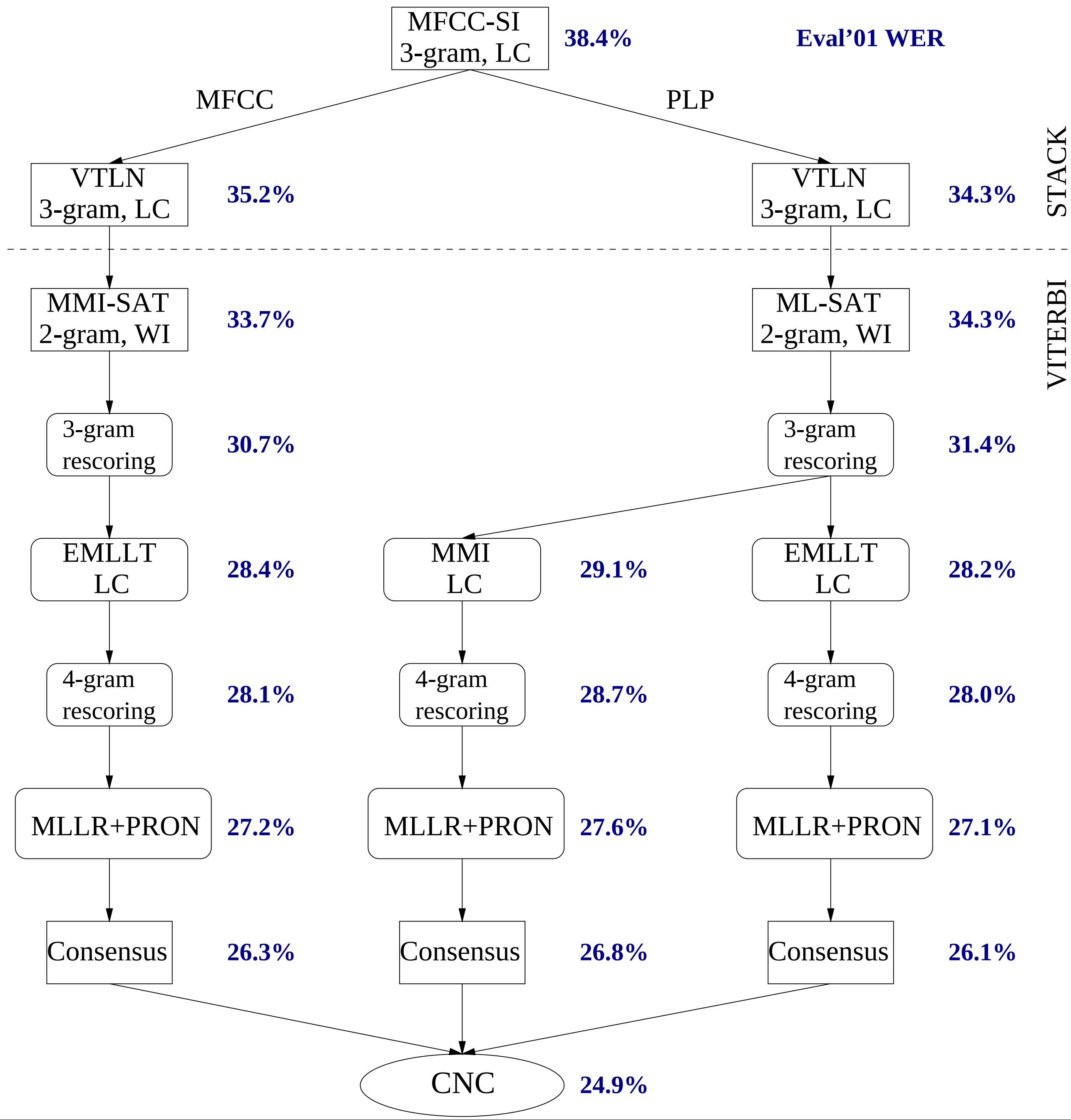
Moved from multi-pass stack decoding to Viterbi lattice generation and rescoring

1. Lattices generated at the SAT+FMLLR level using word-internal AM and 2-gram LM

2. Expanded to 3-grams and left cross-word acoustic context and pruned

3. Rescored and pruned with progressively more accurate models (4-gram LM, lattice-MLLR adapted AM)

4. Turned into confusion networks and combined


MFCC-SI
3-gram, LC
38.4%
Eval'01 WER
MFCC
PLP
VTLN
3-gram, LC
35.2%
VTLN
3-gram, LC
34.3%
STACK
VITERBI
MMI-SAT
2-gram, WI
33.7%
ML-SAT
2-gram, WI
34.3%
3-gram
rescoring
30.7%
3-gram
rescoring
31.4%
EMLLT
LC
28.4%
MMI
LC
29.1%
EMLLT
LC
28.2%
4-gram
rescoring
28.1%
4-gram
rescoring
28.7%
4-gram
rescoring
28.0%
MLLR+PRON
27.2%
MLLR+PRON
27.6%
MLLR+PRON
27.1%
Consensus
26.3%
Consensus
26.8%
Consensus
26.1%
CNC
24.9%

CDF matching adaptation

Introduced by [Dharanipragada & Padmanabhan'00]

- Distribution function (or CDF) of a continuous r.v. X:

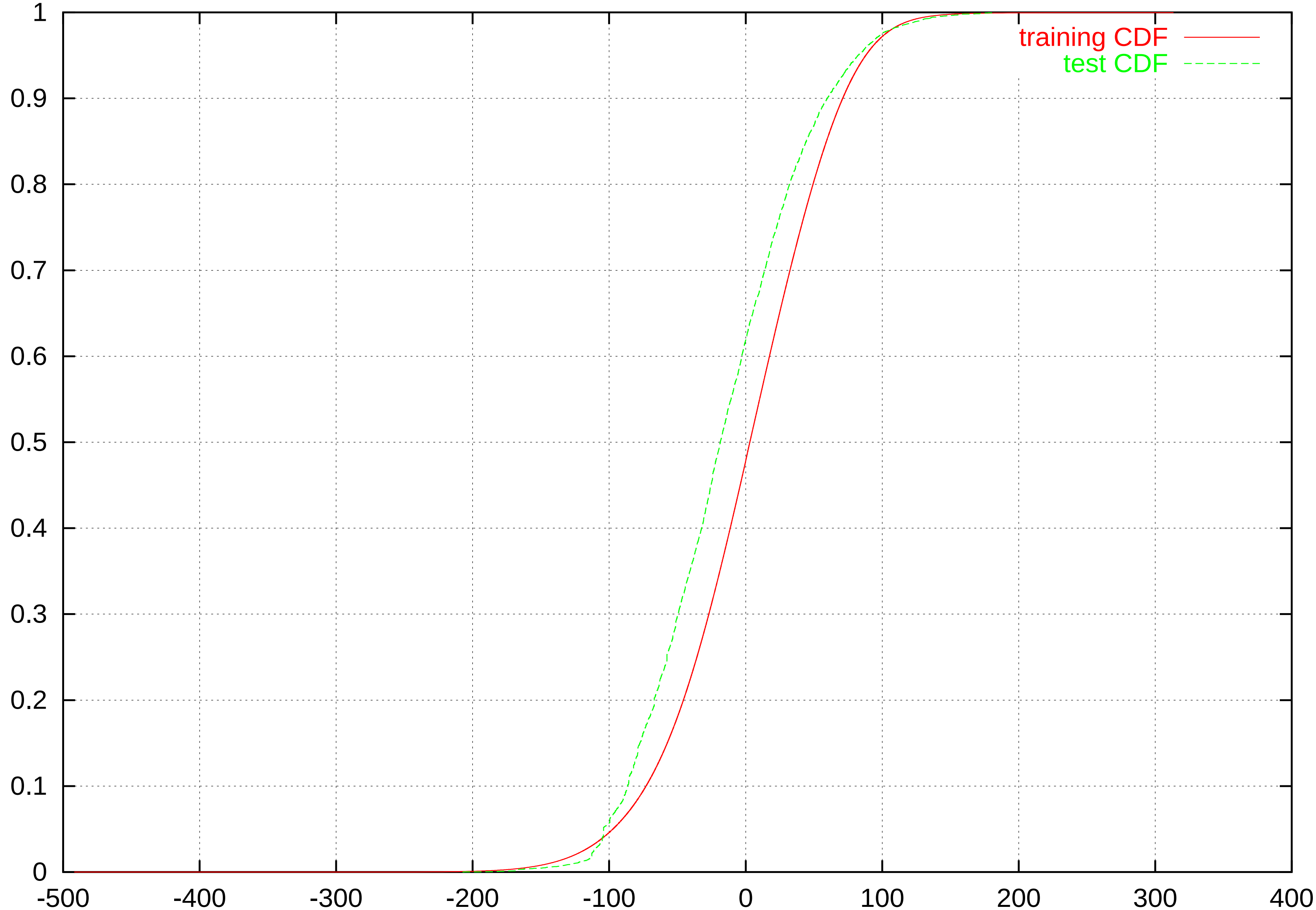
$$F(x) = P(X \leq x) = \int_{-\infty}^{x} p(t)dt$$

- Empirical CDF given training samples $x_1, \ldots, x_N$:

$$F_N(x) = \frac{1}{N}\sum_{i=1}^{N} \theta(x - x_i)$$

- Idea: match the empirical test CDF to the empirical training CDF for each dimension independently

- Related to the Gaussianization technique [Chen & Gopinath'00]


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-500
-400
-300
-200
-100
0
100
200
300
400
training CDF
test CDF

CDF matching adaptation (cont'd)

- Remark: $F_N(x_i) = \dfrac{rank(x_i)}{N}$
- $\mathcal{T} = \{x_1, \ldots, x_N\}$ training data, F_N empirical training CDF
- $\mathcal{A} = \{y_1, \ldots, y_M\}$ adaptation data, G_M empirical test CDF
- mapping $h : \mathcal{A} \to \mathcal{T}$, $h = F_N^{-1} \circ G_M$. Then:

$$F_N(h(y_i)) = G_M(y_i), \qquad \forall y_i \in \mathcal{A}$$

1. Sort the training data
2. Sort the test data
3. Replace each test sample y_i with the training sample $h(y_i)$
4. Decode training data !!!

Decoding results

- Stack decoding:

Model/Transform	eval'00	eval'98	devset cellular
SAT+FMLLR	24.6%	37.7%	39.9%
SAT+FMLLR+FV	24.4%	37.5%	N/A
SAT+FMLLR+CDF+FV	24.6%	N/A	N/A
SAT+FMLLR+CDF+FMLLR	24.4%	37.2%	39.4%

- Lattice rescoring:

Model/Transform	eval'00	eval'98
SAT+FMLLR	23.7%	36.6%
SAT+FMLLR+CDF+FMLLR	23.3%	36.1%

Extended maximum likelihod linear transforms (EMLLT)

Introduced by [Olsen & Gopinath'02]

Idea: model Gaussian precision matrices (inverse covariances) as

$$\mathbf{P}_i = \mathbf{A}\mathbf{\Lambda}_i\mathbf{A}^T$$

where

$$\mathbf{P}_i = \mathbf{\Sigma}_i^{-1} \in \mathbb{R}^{n\times n},\ \mathbf{A} \in \mathbb{R}^{n\times N},\ \mathbf{\Lambda}_i \in \mathbb{R}^{N\times N},\ \mathbf{\Lambda}_i = \mathrm{diag}(\lambda_{i1}\ldots\lambda_{iN})$$

and $n \leq N \leq n(n+1)/2$

- MLLT: $N = n$
- Full-covariance: $N = n(n+1)/2$

Decoding results

Courtesy of [Huang, Goel, Gopinath, Kingsbury, Olsen, Visweswariah'02]

- Stack decoding swb'00 (MFCC features):

Model/Transform	Diagonal	EMLLT
VTLN	26.8%	25.2%
SAT+FMLLR	24.6%	23.1%
SAT+FMLLR+MLLR	23.6%	22.6%

- Lattice rescoring eval'01 (PLP features):

Model/Transform	Diagonal	EMLLT
SAT+FMLLR	29.1%	28.4%
SAT+FMLLR+4grm+MLLR	28.0%	27.2%

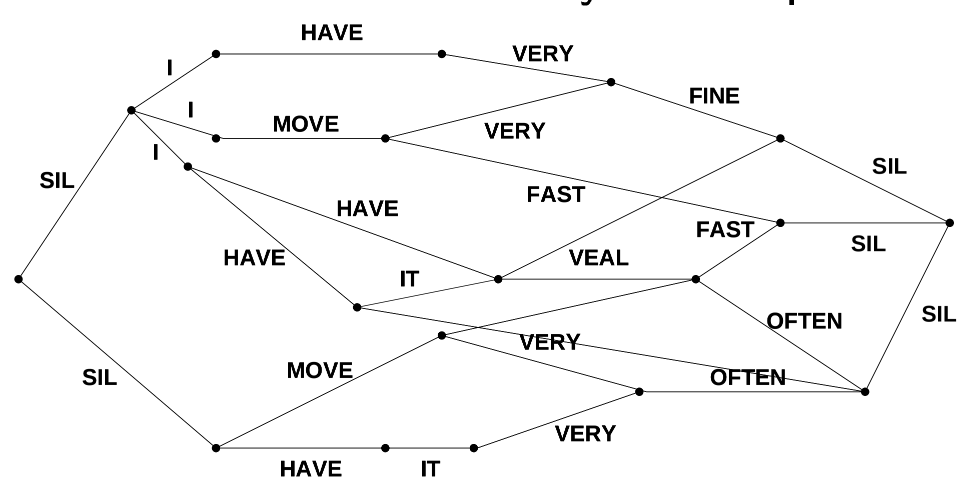
Implicit lattice MMI training

- MMI objective function:

$$f(\lambda) = \sum_{k=1}^{K} \log \frac{P_\lambda(\mathbf{X}^k|\mathbf{W}^k)}{\sum_{\mathbf{W}} P_\lambda(\mathbf{X}^k|\mathbf{W})P(\mathbf{W})}$$

where λ represents the means, variances and priors of the Gaussians

- Compute the denominator statistics only for the paths existent in a lattice



Implicit lattice MMI training (cont'd)

- Previous approach:
  - Create lattice using simpler models (e.g. x-word triphones, or word-internal)
  - Expand lattice to larger acoustic context (x-word quinphones, or left-context) and run Forward-Backward algorithm to accumulate counts
- Proposed method:
  - Statically compile left-context, n-gram decoding graphs: arc minimization problem addressed in [Zweig, Saon & Yvon'02]
  - Run Forward-Backward with pruning (instead of Viterbi) on the resulting HMM network

Decoding results

- Trigram one-shot Viterbi decoding:

Context	Training	eval'00
word-internal	ML	26.1%
	MMI	24.9%
left	ML	25.3%
	MMI	24.0%

- Bigram lattice generation (1-best results):

Context	Training	eval'00
word-internal	ML	27.7%
	MMI	25.8%

Conclusion

Search	=	5% relative improvement
CDF matching adaptation	=	1-2% relative improvement
EMLLT	=	5% relative improvement
Implicit lattice MMI	=	5-7% relative improvement